News Recommender System

Anonymous Alligators
April 20, 2022

Bhavik (MS18098), Aniket (MS18126), Kunal (MS18148), Akshay (MS18117)

Contents

1 Introduction **1**

2 Data Scraping **1**

3 Extracting article features **2**
3.1 Topic Modelling 2
3.1.1 Determining optimal no. of topics 2
3.2 Other feature extraction 3

4 Flask API - User side interface **3**

5 User Profiles **4**
5.1 Gathering user analytics from API 4
5.2 Engagement Scores 4
5.3 Building the profile 5

6 Article Recommendation **5**
6.1 New-User recommendation: Tackling cold-start 5
6.2 Old-User recommendation 6
6.2.1 Item-based recommendation 6
6.2.2 Content-based recommendation 6
6.2.3 Collaborative-based recommendation 6

7 Things we had planned but could not implement **6**

8 Code repository **7**

1 Introduction

In this assignment, we aimed to design, implement and analyze a news recommendation system for the company **Alligator's Den**. The objective was to increase retention of the customers by maximising engagement (through intelligent recommendation of articles), right from their first visit.

2 Data Scraping

We have used the following four websites to scrape the article data for this project (using `BeautifulSoup4` and `requests`): (Republic World, IBTimes, India Today, El Pais)

For each article, we collect a 6-tuple of meta-data like so:

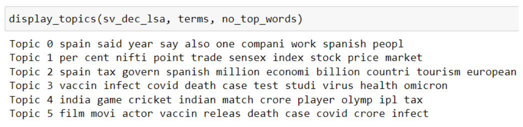
(Headline, Published Date, URL, Category, Summary, Content)

These are then stored in an SQLite database for further pre-processing (duplicate removal, cleaning, lemmatization, stemming, etc) before they are vectorized for further usage. A total of 6000 articles were scraped over the following categories: **Science, Sports, Business, Culture and Entertainment**.

3 Extracting article features

3.1 Topic Modelling

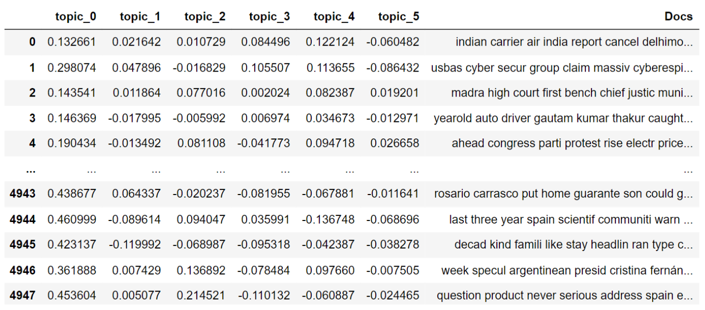
The corpus is vectorized using TF-IDF and fed into an LSA (Latent-Semantic-Analysis) model using the `sklearn` library to extract a number of latent topics from the vectorized corpus. We chose LSA over LDA primarily because topic modelling in LSA was much faster than LDA, but also because we did not find the opportunity to properly tweak the parameters in the LDA model.



```
display_topics(sv_dec_lsa, terms, no_top_words)
```

```
Topic 0 spain said year say also one compani work spanish peopl
Topic 1 per cent nifti point trade sensex index stock price market
Topic 2 spain tax govern spanish million economi billion countri tourism european
Topic 3 vaccin infect covid death case test studi virus health omicron
Topic 4 india game cricket indian match crore player olymp ipl tax
Topic 5 film movi actor vaccin releas death case covid crore infect
```

(a) Latent topics extracted from LSA modelling.



	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	Docs
0	0.132661	0.021642	0.010729	0.084496	0.122124	-0.060482	indian carrier air india report cancel delhimo...
1	0.298074	0.047896	-0.016829	0.105507	0.113655	-0.086432	usbas cyber secur group claim massiv cyberespi...
2	0.143541	0.011864	0.077016	0.002024	0.082387	0.019201	madra high court first bench chief justic muni...
3	0.146369	-0.017995	-0.005992	0.006974	0.034673	-0.012971	yearold auto driver gautam kumar thakur caught...
4	0.190434	-0.013492	0.081108	-0.041773	0.094718	0.026658	ahead congress parti protest rise electr price...
...	...	...	...	...	...	...	...
4943	0.438677	0.064337	-0.020237	-0.081955	-0.067881	-0.011641	rosario carrasco put home guarante son could g...
4944	0.460999	-0.089614	0.094047	0.035991	-0.136748	-0.068696	last three year spain scientif communiti warn ...
4945	0.423137	-0.119992	-0.068987	-0.095318	-0.042387	-0.038278	decad kind famili like stay headlin ran type c...
4946	0.361888	0.007429	0.136892	-0.078484	0.097660	-0.007505	week specul argentinean presid cristina fernán...
4947	0.453604	0.005077	0.214521	-0.110132	-0.060887	-0.024465	question product never serious address spain e...

(b) Articles vs. latent topics

Figure 1: Using Latent Semantic Analysis for topic modelling.

3.1.1 Determining optimal no. of topics

To determine optimal number of topics/components in our LSA model, we established a topic coherence score which uses an average/median of pairwise word similarity scores of the words that are present in the topic. The number of topics with highest coherence score is chosen.

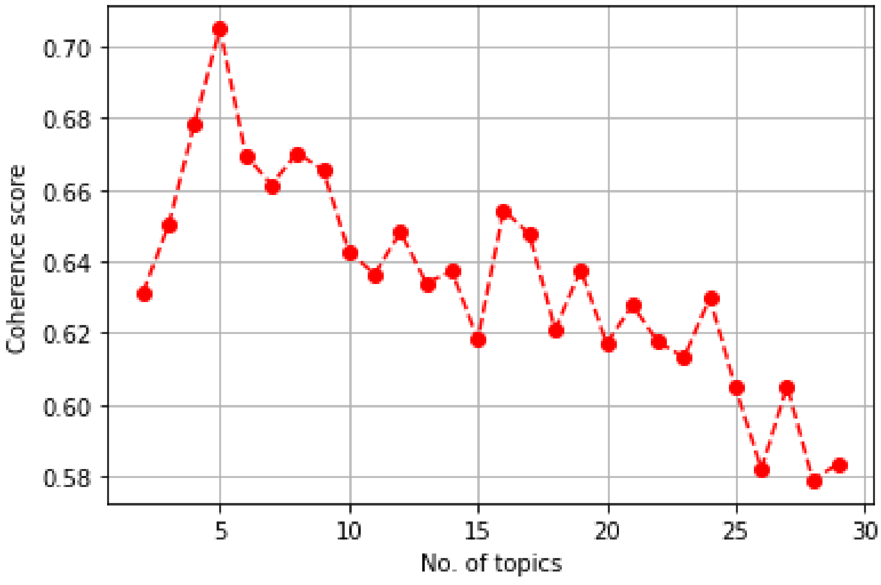


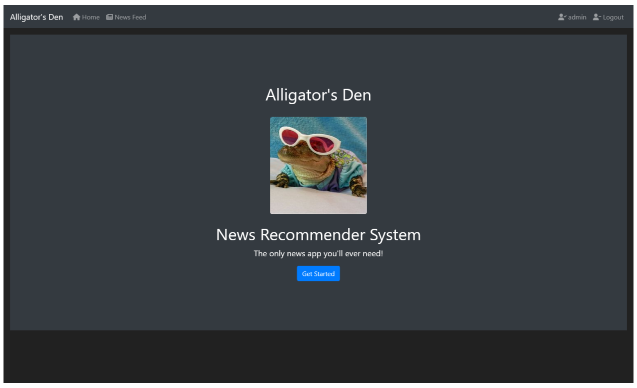
Figure 2: extracting coherence score from LSA model

Therefore, to extract optimal number of topics we performed LSA for a range of topics, from which coherence scores were calculated using `gensim`. The model gave us the highest coherence score for 5 topics as shown in the plot above, which roughly matches the number of news categories we scraped from (six).

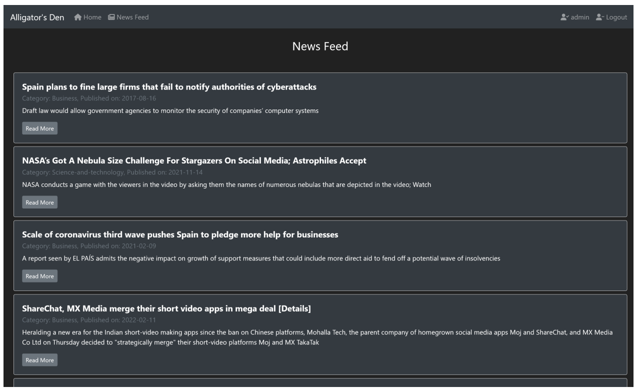
3.2 Other feature extraction

We also tried using Named-Entity-Recognition (to specifically obtain contextual keywords, instead of random words) as a secondary filter in addition to preliminary similarity comparisons. However, this gave redundant results to that of the LSA modelling, so we proceeded without it.

4 Flask API - User side interface



(a) Home Page



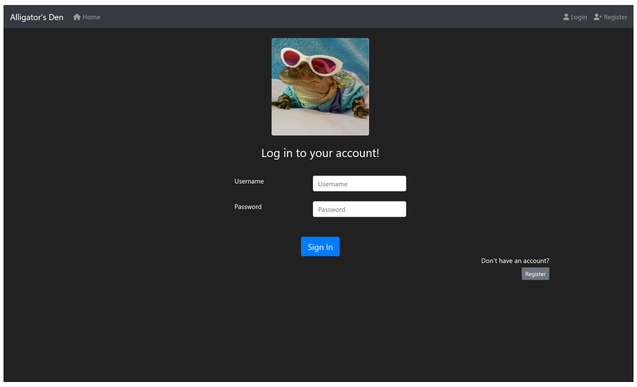
Spain plans to fine large firms that fail to notify authorities of cyberattacks
Category: Business, Published on: 2017-08-16
Draft law would allow government agencies to monitor the security of companies' computer systems
Read More

NASA's Got A Nebula Size Challenge For Stargazers On Social Media; Astrophiles Accept
Category: Science-and-technology, Published on: 2021-11-14
NASA conducts a game with the viewers in the video by asking them the names of numerous nebulas that are depicted in the video; Watch
Read More

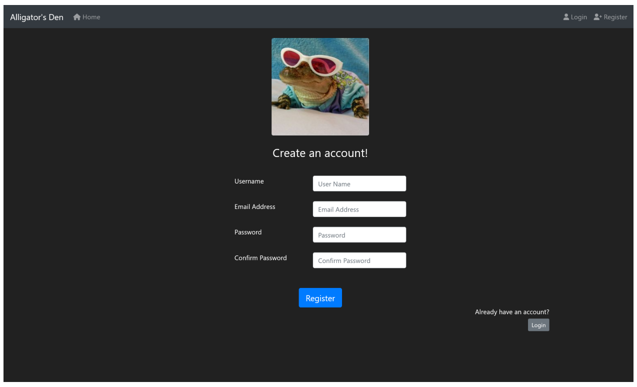
Scale of coronavirus third wave pushes Spain to pledge more help for businesses
Category: Business, Published on: 2021-02-09
A report seen by EL PAÍS admits the negative impact on growth of support measures that could include more direct aid to fend off a potential wave of insolvencies
Read More

ShareChat, MX Media merge their short video apps in mega deal [Details]
Category: Business, Published on: 2022-02-11
Heralding a new era for the Indian short-video making apps since the ban on Chinese platforms, Mohalla Tech, the parent company of homegrown social media apps Moj and ShareChat, and MX Media Co Ltd on Thursday decided to "strategically merge" their short-video platforms Moj and MX TakaTak
Read More

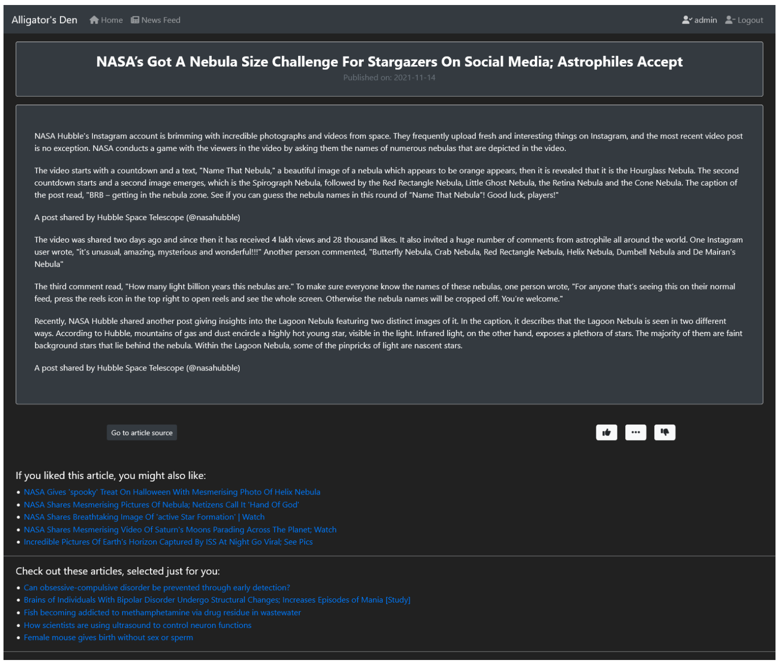
(b) News Feed



(c) Login Page



(d) Register Page



NASA's Got A Nebula Size Challenge For Stargazers On Social Media; Astrophiles Accept
Published on: 2021-11-14

NASA Hubble's Instagram account is brimming with incredible photographs and videos from space. They frequently upload fresh and interesting things on Instagram, and the most recent video post is no exception. NASA conducts a game with the viewers in the video by asking them the names of numerous nebulas that are depicted in the video.

The video starts with a countdown and a text, "Name That Nebula," a beautiful image of a nebula which appears to be orange appears, then it is revealed that it is the Hourglass Nebula. The second countdown starts and a second image emerges, which is the Spirograph Nebula, followed by the Red Rectangle Nebula, Little Ghost Nebula, the Retina Nebula and the Cone Nebula. The caption of the post read, "BRB – getting in the nebula zone. See if you can guess the nebula names in this round of "Name That Nebula"! Good luck, players!"

A post shared by Hubble Space Telescope (@nasahubble)

The video was shared two days ago and since then it has received 4 lakh views and 28 thousand likes. It also invited a huge number of comments from astrophile all around the world. One Instagram user wrote, "it's unusual, amazing, mysterious and wonderful!!!" Another person commented, "Butterfly Nebula, Crab Nebula, Red Rectangle Nebula, Helix Nebula, Dumbell Nebula and De Mairan's Nebula"

The third comment read, "How many light billion years this nebulas are." To make sure everyone know the names of these nebulas, one person wrote, "For anyone that's seeing this on their normal feed, press the reels icon in the top right to open reels and see the whole screen. Otherwise the nebula names will be cropped off. You're welcome."

Recently, NASA Hubble shared another post giving insights into the Lagoon Nebula featuring two distinct images of it. In the caption, it describes that the Lagoon Nebula is seen in two different ways. According to Hubble, mountains of gas and dust encircle a highly hot young star, visible in the light. Infrared light, on the other hand, exposes a plethora of stars. The majority of them are faint background stars that lie behind the nebula. Within the Lagoon Nebula, some of the pinpricks of light are nascent stars.

A post shared by Hubble Space Telescope (@nasahubble)

If you liked this article, you might also like:
- NASA Gives 'spooky' Treat On Halloween With Mesmerising Photo Of Helix Nebula
- NASA Shares Mesmerising Pictures Of Nebula; Netizens Call It 'Hand Of God'
- NASA Shares Breathtaking Image Of 'active Star Formation' | Watch
- NASA Shares Mesmerising Video Of Saturn's Moons Parading Across The Planet; Watch
- Incredible Pictures Of Earth's Horizon Captured By ISS At Night Go Viral; See Pics

Check out these articles, selected just for you:
- Can obsessive-compulsive disorder be prevented through early detection?
- Brains of Individuals With Bipolar Disorder Undergo Structural Changes; Increases Episodes of Mania [Study]
- Fish becoming addicted to methamphetamine via drug residue in wastewater
- How scientists are using ultrasound to control neuron functions
- Female mouse gives birth without sex or sperm

Figure 4: A preview of the article page

Upon first visit, the user is prompted to register for an account (to facilitate data tracking to build the user profile). Upon login, the user is redirected to the News Feed which initially has a list of randomly chosen articles from the user's preferred categories. This section continues to display a randomized selection of articles to encourage exploration. The actual recommendations (personalized) happen in individual article pages, in two sections; (1) similar articles to the current one, (2) content-based + collab-based suggestions using user data. Every article page also has 3 (optional) explicit rating buttons that will be used to generate engagement scores.

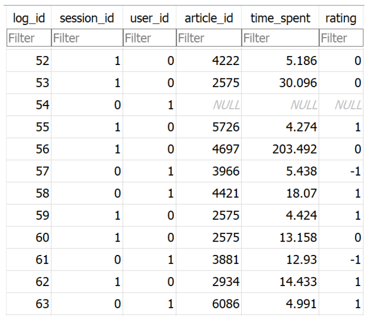
5 User Profiles

In order to make recommendations, we have gathered some analytics from each user session to extract information that is aggregated to calculate an 'engagement score' ($e \in [0, 1]$) for each document that a user visits.

5.1 Gathering user analytics from API

Since we have reconstructed the article pages using scraped data, we utilize a small javascript snippet to track the time spent by the user on the page. Further, we have an explicit rating system ($r \in \{-1, 0, 1\}$) that the user can optionally provide for each article they read.

When the page is unloaded (another link is clicked), the gathered information is sent to the Flask server as a `POST` request using `navigator.beacon(...)` to the `/analytics` route, which is then stored in the SQLite database.



log_id	session_id	user_id	article_id	time_spent	rating
Filter	Filter	Filter	Filter	Filter	Filter
52	1	0	4222	5.186	0
53	1	0	2575	30.096	0
54	0	1	NULL	NULL	NULL
55	1	0	5726	4.274	1
56	1	0	4697	203.492	0
57	0	1	3966	5.438	-1
58	0	1	4421	18.07	1
59	1	0	2575	4.424	1
60	1	0	2575	13.158	0
61	0	1	3881	12.93	-1
62	1	0	2934	14.433	1
63	0	1	6086	4.991	1

5.2 Engagement Scores

We have restricted our engagement scores to take any value in the range $\{-1\} \cup [0, 1]$. As seen above, the flask API gathers data about explicit ratings and time spent on the article page. If the user provides an explicit rating (one of {-1, 0, 1}), this is assigned as the engagement score, interpreted in the following way.

- -1 $\rightarrow$ The user **dislikes** the article
- 0 $\rightarrow$ The user did not read the article and hence is **uninterested/indifferent** to it.
- 1 $\rightarrow$ The user **likes** the article.

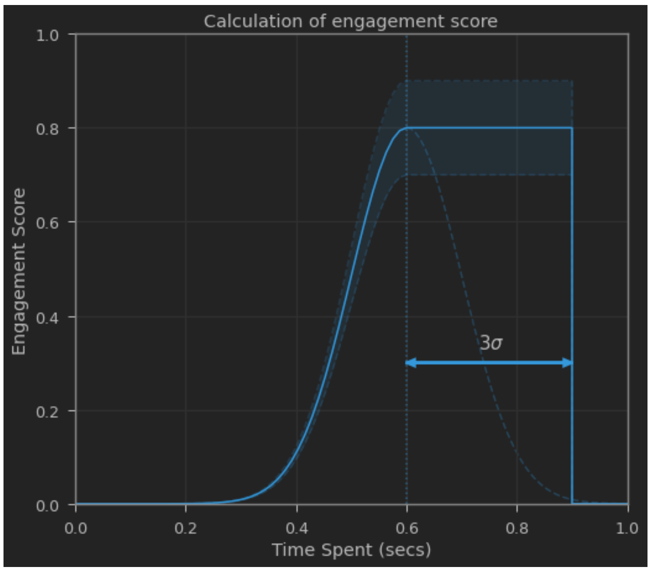


Figure 5: Calculation of the implicit engagement score (normalized X-axis)

In the absence of an explicit rating, we utilize the following scheme to generate an implicit rating using the time spent on the page. The distribution of reading times for an article, d_i is modelled as a gaussian characterized by :

$$\mu = (\#\text{ words in } d_i)/\mu_w \qquad \sigma = (\#\text{ words in } d_i)/\sigma_w$$

where μ_w is the number of words that an average user reads per second. The following values were utilized from this study [1]: $(\mu_w, \sigma_w) = (247\text{ wpm}, 46\text{ wpm}) \equiv (4.11\text{ wps}, 0.76\text{ wps})$.
It should be noted that our implicit rating scheme generates a rating within the range [0, 1] and not [-1, 1]. This is because, given only the time spent on the page, it only makes sense to deduce the level of interest/*engagement*, and not the level of dislike for the article.

The engagement score is simply taken to be proportional to this reading time curve (re-scaled to $[0, 1]$), the logic being that the rating gets higher as the mean reading time is reached. After the user crosses the mean reading time, the rating stagnates at a maximum value, e_{max} till $t < \mu + 3\sigma$, after which the rating drops to 0. This is done because after a certain period, the user is likely to have lost interest and left the page without closing it. As a result, this time information is useless to deduce user interests and is treated as if the user did not read the article at all.

Finally, even if a user reads an article exactly in the time region $\mu < t < \mu + 3\sigma$, we cannot assign a rating of 1 as given explicitly by a user who has clicked the +1 button, since this is simply an estimation. Instead, we randomly sample the maximum rating, e_{max} (peak of gaussian) from a uniform distribution between 0.85 and 0.95.

5.3 Building the profile

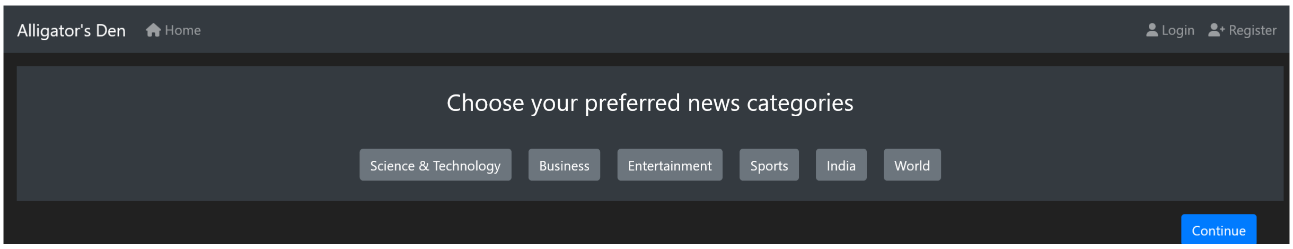
As part of the click-stream data, we possess a list of all documents ($\{\vec{d_i}\}$) that a user has visited, as well the engagement scores ($\{e_i\}$). A user is simply taken to be a linear combination of the documents they have visited, with the weights being the engagement scores:

$$\vec{u_j} = \frac{\sum_i e_{ji} \cdot \vec{d_i}}{||\sum_i e_{ji} \cdot \vec{d_i}||} \qquad \text{s.t. } i \in \{\text{articles consumed by the user}\}$$

The user vector $\vec{u_j}$ is normalized for consistency, as the cosine similarity metric is agnostic to the magnitude of the vectors anyways. The document vectors, $\vec{d_i}$ used here are not a bag of words, but rather a bag of topics (retrieved from LSA). The idea is that, taking into account the topics rather than words will retain more context about user preferences upon taking a linear combination.

6 Article Recommendation

6.1 New-User recommendation: Tackling cold-start



- Upon registration and first login, a prompt of 6 prominent news categories is presented that the user chooses from (knowledge-based recommendation). This data is then used to present a randomized selection from the relevant part of the corpus in the news-feed (expected to be more similar to the user's tastes). This persists for the first 5 user sessions, and then the category preferences are forgotten to encourage exploration.
- After visiting 5 articles, the content-based recommendation system kicks in, and soon after, a hybrid approach (with collaborative filtering) is utilized.

6.2 Old-User recommendation

We will utilize a hybrid approach of content-based and collaborative filtering, wherein the top 5 articles from each of the methods are generated and presented to the user.

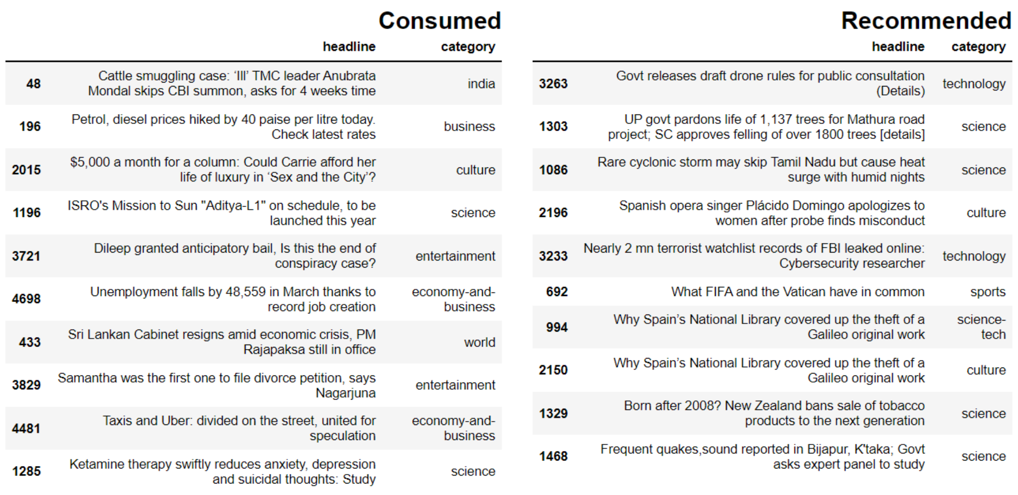
6.2.1 Item-based recommendation

In order to gather some initial data by a naive method, we simply recommend the 5 nearest neighbour articles to the article that is being currently read. This does not take into account the user's preferences.

6.2.2 Content-based recommendation

Our content-based recommender is based on the principle of finding the set of nearest articles to the user $\vec{u_j}$. We chose our similarity metric to be the cosine similarity -

$$s(\vec{u_j}, \vec{d_k}) = \frac{\vec{u_j} \cdot \vec{d_k}}{||\vec{u_j}|| \; ||\vec{d_k}||}$$



Consumed

	headline	category
48	Cattle smuggling case: 'Ill' TMC leader Anubrata Mondal skips CBI summon, asks for 4 weeks time	india
196	Petrol, diesel prices hiked by 40 paise per litre today. Check latest rates	business
2015	$5,000 a month for a column: Could Carrie afford her life of luxury in 'Sex and the City'?	culture
1196	ISRO's Mission to Sun "Aditya-L1" on schedule, to be launched this year	science
3721	Dileep granted anticipatory bail, Is this the end of conspiracy case?	entertainment
4698	Unemployment falls by 48,559 in March thanks to record job creation	economy-and-business
433	Sri Lankan Cabinet resigns amid economic crisis, PM Rajapaksa still in office	world
3829	Samantha was the first one to file divorce petition, says Nagarjuna	entertainment
4481	Taxis and Uber: divided on the street, united for speculation	economy-and-business
1285	Ketamine therapy swiftly reduces anxiety, depression and suicidal thoughts: Study	science

Recommended

	headline	category
3263	Govt releases draft drone rules for public consultation (Details)	technology
1303	UP govt pardons life of 1,137 trees for Mathura road project; SC approves felling of over 1800 trees [details]	science
1086	Rare cyclonic storm may skip Tamil Nadu but cause heat surge with humid nights	science
2196	Spanish opera singer Plácido Domingo apologizes to women after probe finds misconduct	culture
3233	Nearly 2 mn terrorist watchlist records of FBI leaked online: Cybersecurity researcher	technology
692	What FIFA and the Vatican have in common	sports
994	Why Spain's National Library covered up the theft of a Galileo original work	science-tech
2150	Why Spain's National Library covered up the theft of a Galileo original work	culture
1329	Born after 2008? New Zealand bans sale of tobacco products to the next generation	science
1468	Frequent quakes,sound reported in Bijapur, K'taka; Govt asks expert panel to study	science

Figure 6: 10 content-based recommendations from a user who has read 10 articles.

6.2.3 Collaborative-based recommendation

For the collaborative system, we focus on the *user-based implementation*, which involves recommending articles based on the interests of similar users. A brief outline of the process is as follows:

1. Find the nearest n-neighbours to a given user.
2. Rate all of the articles in the corpus by taking the average of the ratings given by these n-neighbouring users.
3. Rank the next top 5 recommendations (with which the user hasn't interacted before) based on descending average ratings computed from the above step.

Since majority of the rating matrix is filled with `nan` values (since the corpus size is huge and most articles are unread), we haven't displayed it here.

7 Things we had planned but could not implement

There were several other things that we had planned but we couldn't due to the lack of expertise and time. Some of them are as follows:

- We had an issue of redundant recommendations where a user who has read a particular article is recommended that article again. While removing the read articles from recommendations is easy, we wanted to incorporate a more elaborate scheme of negatively weighting those articles, so they still have a chance to be recommended again (like in YouTube videos).
- In our item-based recommendation (where the articles similar to the current article are recommended), we faced a cyclic recommendation issue, wherein the recommendations all had each other as their nearest neighbours, so it created a loop that the user gets stuck into. We could fix it by introducing some randomness in the recommendation.
- As of now, our corpus size is fixed which means we can't always have the latest news on our website. We wished to make a corpus which would dynamically update itself but it caused certain issues due to which we weren't able to implement it.
- We couldn't spend enough time playing around with LDA parameters to make it more efficient (atleast comparable to LSA).
- Instead of simulating the initial set of users and their activity, we manually went through and read news articles trying to emulate user behaviour because we weren't able to find a good enough model to simulate our data.
- We wanted to add a time-based decay factor to the article rating scheme during recommendation to take into account the "freshness" of the news. But we did not have time to implement/play around with this.
- For collaborative-filtering, we intended to utilize the matrix factorization method but the sparsity of our ratings matrix caused long run-times and bad results. It is entirely possible that we simply did not have enough data, or did not tweak the parameters properly. Given more time, this could be done.

8 Code repository

This project was written in Python 3.9.7 utilizing several data-science and web framework libraries, so we have documented our progress in the git repository linked below. All the code for the final API, along with the jupyter notebooks we used to experiment can be found in the following github link:

`https://github.com/20akshay00/News-Recommender-System/`

Go through the README.md file for instructions to setup the environment and run the project locally.

References

[1] Brysbaert, Marc. 2019. *"How Many Words Do We Read Per Minute? A Review and Meta-analysis of Reading Rate."* PsyArXiv. April 12. doi:10.31234/osf.io/xynwg.